Lab Manual

for

Linear Algebra

by

Jim Hefferon

Cover: my Chocolate Lab, Suzy.

Contents

Maps

We've used Sage to define vector spaces. Next we explore operations that you can do on vector spaces.

Left/right

Sage represents linear maps differently than the book does. An example is in representing the application of a linear map to a member of a vector space $t(\vec{v})$. Consider this function $t\colon \mathbb{R}^2 \to \mathbb{R}^3$ and this element of the domain

$$t(\begin{pmatrix} a \\ b \end{pmatrix}) = \begin{pmatrix} a+b \\ a-b \\ b \end{pmatrix} \qquad \vec{v} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

the map application gives this.

$$t(\begin{pmatrix} 1 \\ 3 \end{pmatrix}) = \begin{pmatrix} 4 \\ -2 \\ 3 \end{pmatrix}$$

To represent the map application we first fix bases. In this example we use the canonical bases $E_2 \subset \mathbb{R}^2$ and $E_3 \subset \mathbb{R}^3$. Then with respect to the bases the book finds a matrix $T = \mathrm{Rep}_{E_2,E_3}(t)$ and a column vector $\vec{w} = \mathrm{Rep}_{E_2}(\vec{v})$, and represents $t(\vec{v})$ with the product $T\vec{w}$.

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ -2 \\ 3 \end{pmatrix}$$

That is, the book is write right: its notation puts the vector on the right of the matrix.

However, this choice is a matter of taste and many authors instead use a row vector that multiplies a matrix from the left. Sage is in this camp and represents the map application in this way.

$$\begin{pmatrix} 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 4 & -2 & 3 \end{pmatrix}$$

Obviously the difference is cosmetic but can cause confusion. Rather than quarrel with the tool, in this manual we will do it Sage's way. The translation is that, compared to the book's representation $T\vec{w}$, Sage prefers the transpose $(T\vec{w})^{\mathsf{T}} = \vec{w}^{\mathsf{T}}\, T^{\mathsf{T}}$.

Defining

We will see two different ways to define a linear transformation.

Symbolically We first define a map that takes two inputs and returns three outputs.

```
sage: a, b = var('a, b')
sage: T_symbolic(a, b) = [a+b, a-b, b]
sage: T_symbolic
(a, b) |--> (a + b, a - b, b)
```

We have not yet defined a domain and codomain so this not a function — instead it is a prototype for a function. Make an instance of a function by applying $T_{symbolic}$ on a particular domain and codomain.

```
sage: T = linear_transformation(QQ^2, QQ^3, T_symbolic)
sage: T
Vector space morphism represented by the matrix:
[ 1  1  0]
[ 1 -1  1]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 3 over Rational Field
```

Note the left/right issue again: Sage's matrix is the transpose of the matrix that the book would use.

Evaluating this function on a member of the domain gives a member of the codomain.

```
sage: v = vector(QQ, [1, 3])
sage: v
(1, 3)
sage: T(v)
(4, -2, 3)
```

Sage can compute the interesting things about the transformation. Here it finds the null space and range space, using the equivalent terms *kernel* and *image*.

```
sage: T.kernel()
Vector space of degree 2 and dimension 0 over Rational Field
Basis matrix:
[]
sage: T.image()
Vector space of degree 3 and dimension 2 over Rational Field
Basis matrix:
[   1    0  1/2]
[   0    1 -1/2]
```

The null space's basis is empty because it is the trivial subspace of the domain, with dimension 0. Therefore T is one-to-one.

The range space has a 2-vector basis. This agrees with the theorem that the dimension of the null space plus the dimension of the range space equals to the dimension of the domain.

For contrast consider a map that is not one-to-one.

```
sage: S_symbolic(a, b) = [a+2*b, a+2*b]
sage: S_symbolic
(a, b) |--> (a + 2*b, a + 2*b)
```

```
sage: S = linear_transformation(QQ^2, QQ^2, S_symbolic)
sage: S
Vector space morphism represented by the matrix:
[1 1]
[2 2]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: v = vector(QQ, [1, 3])
sage: S(v)
(7, 7)
```

This map is not one-to-one since the input $(a, b) = (2, 0)$ gives the same result as $(a, b) = (0, 1)$.

```
sage: S(vector(QQ, [2, 0]))
(2, 2)
sage: S(vector(QQ, [0, 1]))
(2, 2)
```

Another way to tell that the map is not one-to-one is to look at the nullspace.

```
sage: S.kernel()
Vector space of degree 2 and dimension 1 over Rational Field
Basis matrix:
[   1 -1/2]
sage: S.image()
Vector space of degree 2 and dimension 1 over Rational Field
Basis matrix:
[1 1]
```

The null space has nonzero dimension, namely it has dimension 1, so Sage agrees that the map is not one-to-one.

Without looking at the range space we know that its dimension must be 1 because the dimensions of the null and range spaces add to the dimension of the domain. Again, Sage confirms our calculation.

Via matrices We can define a transformation by specifying the matrix representing its action.

```
sage: M = matrix(QQ, [[1, 2], [3, 4], [5, 6]])
sage: M
[1 2]
[3 4]
[5 6]
sage: m = linear_transformation(M)
sage: m
Vector space morphism represented by the matrix:
[1 2]
[3 4]
[5 6]
Domain: Vector space of dimension 3 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
```

Note again that Sage prefers the representation where the vector multiplies from the left.

```
sage: v = vector(QQ, [7, 8, 9])
sage: v
```

```
(7, 8, 9)
sage: m(v)
(76, 100)
```

Sage has done this calculation.

$$\begin{pmatrix} 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} = \begin{pmatrix} 76 & 100 \end{pmatrix}$$

If you have a matrix intended for a vector-on-the-right calculation (as in the book) then Sage will make the necessary adaptation.

```
sage: N = matrix(QQ, [[1, 3, 5], [2, 4, 6]])
sage: n = linear_transformation(N, side='right')
sage: n
Vector space morphism represented by the matrix:
[1 2]
[3 4]
[5 6]
Domain: Vector space of dimension 3 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: v = vector(QQ, [7, 8, 9])
sage: v
(7, 8, 9)
sage: n(v)
(76, 100)
```

Although we gave it a `side='right'` option, the matrix that Sage shows by default is for `side='left'`.

Despite that we specified them differently, these two transformations are the same.

```
sage: M = matrix(QQ, [[1, 2], [3, 4], [5, 6]])
sage: m = linear_transformation(M)
sage: N = matrix(QQ, [[1, 3, 5], [2, 4, 6]])
sage: n = linear_transformation(N, side='right')
sage: m == n
True
```

We can ask the same questions of linear transformations created from matrices that we asked of linear transformations created from functions.

```
sage: M = matrix(QQ, [[1, 2], [3, 4], [5, 6]])
sage: m = linear_transformation(M)
sage: m.kernel()
Vector space of degree 3 and dimension 1 over Rational Field
Basis matrix:
[ 1 -2  1]
```

The null space of m is not the trivial subspace of $\mathbb{R}^3$ and so this function is not one-to-one. The domain has dimension 3 and the null space has dimension 1 and so the range space is a dimension 2 subspace of $\mathbb{R}^2$.

```
sage: m.image()
Vector space of degree 2 and dimension 2 over Rational Field
```

```
Basis matrix:
[1 0]
[0 1]
sage: m.image() == QQ^2
True
```

Sage lets us have the matrix represent a transformation involving spaces with nonstandard bases.

```
sage: M = matrix(QQ, [[1, 2], [3, 4]])
sage: delta_1 = vector(QQ, [1, -1])
sage: delta_2 = vector(QQ, [1, 1])
sage: domain_basis = [delta_1, delta_2]
sage: domain_basis
[(1, -1), (1, 1)]
sage: D = (QQ^2).subspace_with_basis(domain_basis)
sage: gamma_1 = vector(QQ, [2, 0])
sage: gamma_2 = vector(QQ, [0, 3])
sage: codomain_basis = [gamma_1, gamma_2]
sage: codomain_basis
[(2, 0), (0, 3)]
sage: C = (QQ^2).subspace_with_basis(codomain_basis)
sage: m = linear_transformation(D, C, M)
sage: m
Vector space morphism represented by the matrix:
[1 2]
[3 4]
Domain: Vector space of degree 2 and dimension 2 over Rational Field
User basis matrix:
[ 1 -1]
[ 1  1]
Codomain: Vector space of degree 2 and dimension 2 over Rational Field
User basis matrix:
[2 0]
[0 3]
sage: m(vector(QQ, [1, 0]))
(4, 9)
```

Sage has calculated that

$$\begin{pmatrix}1\\0\end{pmatrix} = (1/2)\begin{pmatrix}1\\-1\end{pmatrix} + (1/2)\begin{pmatrix}1\\1\end{pmatrix} \quad \text{so} \quad \mathrm{Rep}_{domain\_basis}(\begin{pmatrix}1\\0\end{pmatrix}) = \begin{pmatrix}1/2\\1/2\end{pmatrix}$$

and then computed this.

$$\mathrm{Rep}_{codomain\_basis}(m(\vec{v})) = \begin{pmatrix}1/2 & 1/2\end{pmatrix}\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} = \begin{pmatrix}2 & 3\end{pmatrix} \quad \text{so} \quad m(\vec{v}) = 2\begin{pmatrix}2\\0\end{pmatrix} + 3\begin{pmatrix}0\\3\end{pmatrix} = \begin{pmatrix}4\\9\end{pmatrix}$$

Operations

Fix some vector space domain D and codomain C and consider the set of all linear transformations between them. This set has some natural operations, including addition and scalar multiplication. Sage can work with those operations.

Addition Recall that matrix addition is defined so that the representation of the sum of two linear transformations is the matrix sum of the representatives. Sage can illustrate.

```
sage: M = matrix(QQ, [[1, 2], [3, 4]])
sage: m = linear_transformation(QQ^2, QQ^2, M)
sage: m
Vector space morphism represented by the matrix:
[1 2]
[3 4]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: N = matrix(QQ, [[5, -1], [0, 7]])
sage: n = linear_transformation(QQ^2, QQ^2, N)
sage: n
Vector space morphism represented by the matrix:
[ 5 -1]
[ 0  7]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: m+n
Vector space morphism represented by the matrix:
[ 6  1]
[ 3 11]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: M+N
[ 6  1]
[ 3 11]
```

Similarly, linear map scalar multiplication is reflected by matrix scalar multiplication.

```
sage: m*3
Vector space morphism represented by the matrix:
[ 3  6]
[ 9 12]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: M*3
[ 3  6]
[ 9 12]
```

Composition The composition of linear maps gives rise to matrix multiplication. Sage uses the star $*$ to denote composition of linear maps.

```
sage: M = matrix(QQ, [[1, 2], [3, 4]])
sage: m = linear_transformation(QQ^2, QQ^2, M)
sage: m
Vector space morphism represented by the matrix:
[1 2]
[3 4]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: N = matrix(QQ, [[5, -1], [0, 7]])
```

```
sage: n = linear_transformation(QQ^2, QQ^2, N)
sage: n
Vector space morphism represented by the matrix:
[ 5 -1]
[ 0  7]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: m*n
Vector space morphism represented by the matrix:
[ 2  6]
[21 28]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
```

Note: there is a left/right issue here. As the book emphasizes, matrix multiplication is about representing the composition of the maps. The composition $m \circ n$ is the map $\vec{v} \mapsto m(n(\vec{v}))$, with n applied first. We can walk through the calculation of applying n first and then m

$$\begin{pmatrix}1\\2\end{pmatrix} \stackrel{n}{\longmapsto} \begin{pmatrix}5\\13\end{pmatrix} \stackrel{m}{\longmapsto} \begin{pmatrix}44\\62\end{pmatrix}$$

via these matrix multiplications.

$$\begin{pmatrix}1 & 2\end{pmatrix}\begin{pmatrix}5 & -1\\0 & 7\end{pmatrix} = \begin{pmatrix}5 & 13\end{pmatrix} \quad \text{followed by} \quad \begin{pmatrix}5 & 13\end{pmatrix}\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} = \begin{pmatrix}44 & 62\end{pmatrix}$$

Sage prefers the representing vector on the left so to be the first one done, N must come left-most: $m \circ n$ is represented by NM. Here Sage does the map application.

```
sage: M = matrix(QQ, [[1, 2], [3, 4]])
sage: N = matrix(QQ, [[5, -1], [0, 7]])
sage: M*N
[ 5 13]
[15 25]
sage: N*M
[ 2  6]
[21 28]
sage: m = linear_transformation(QQ^2, QQ^2, M)
sage: n = linear_transformation(QQ^2, QQ^2, N)
sage: t = m*n
sage: t
Vector space morphism represented by the matrix:
[ 2  6]
[21 28]
Domain: Vector space of dimension 2 over Rational Field
Codomain: Vector space of dimension 2 over Rational Field
sage: v = vector(QQ, [1, 2])
sage: t(v)
(44, 62)
```

Bibliography

Robert A. Beezer. Sage for Linear Algebra. http://linear.ups.edu/download/fcla-2.22-sage-4.7.1-preview.pdf, 2011.

David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, March 1991.

Jim Hefferon. Linear Algebra. http://joshua.smcvt.edu/linearalgebra, 2012.

David Joyner and William Stein. Open source mathematical software. *Notices of the AMS*, page 1279, November 2007.

Python Team. Floating point arithmetic: issues and limitations, 2012a. URL http://docs.python.org/2/tutorial/floatingpoint.html. [Online; accessed 17-Dec-2012].

Python Team. The python tutorial, 2012b. URL http://docs.python.org/2/tutorial/index.html. [Online; accessed 17-Dec-2012].

Ron Brandt. *Powerful Learning*. Association for Supervision and Curriculum Development, 1998.

Sage Development Team. Sage tutorial 5.3. http://www.sagemath.org/pdf/SageTutorial.pdf, 2012a.

Sage Development Team. Sage reference manual 5.3. http://www.sagemath.org/pdf/reference.pdf, 2012b.

Shunryu Suzuki. *Zen Mind, Beginners Mind*. Shambhala, 2006.